AG44 : Ski Resort

Guillaume OBERLE

9 December 2013

1 Introduction

A director of a large ski resort of Savoie seeks to optimize the business management to make the ski area more attractive for the tourist. He wants to install kiosks in the ski queues and various points of the station which allow the skiers to choose their routes. He want to explain them how to travel to any other point of the resort as quickly as possible.

2 Data Structures

The graph is represented by an object. It contains a list of nodes and a list of edges. Each nodes and edges is a tuple in the list and contains a dictionary for the attributes. Here is an example :

An edge (1,2) : (1, 2, {'color': 'grey', 'type': 'TS', 'weight': 600.0, 'label': 'replat'})
The type attribute define the type of the route.
The weight attribute define the time in second to traverse this route.
The label attribute define the name of the route.
The color attribute is used to draw the graph.

A node (1) : (1, {'altitude': 1200, 'label': 'villaroger'})
The altitude attribute define the altitude of the place.
The label attribute define the name of the place.

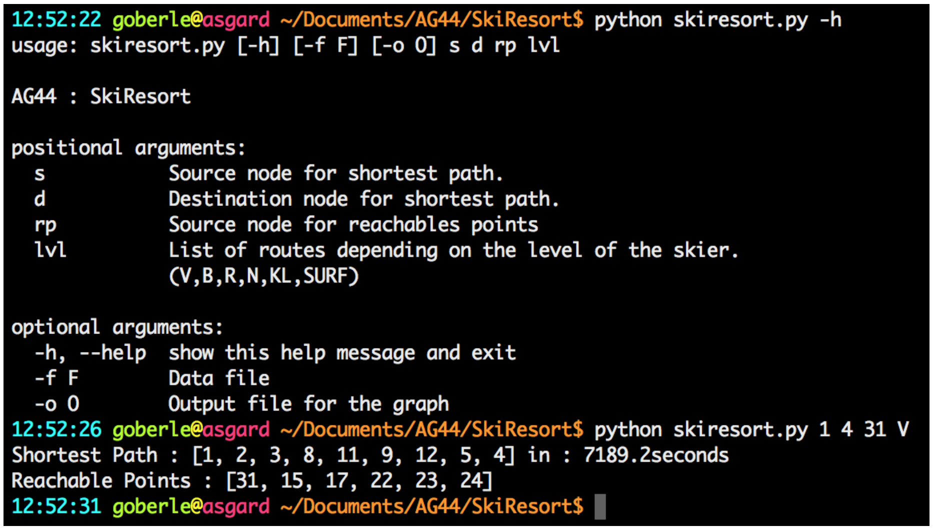
3 Algorithms

To determine the shortest path, I have used the Floyd Warshall algorithm. I choose this algorithm because we have no negative weight and no negative cycle. Moreover, this software may be installed in a kiosk, and I think it's more efficient to calculate the shortest path between all pairs of vertices at the beginning and reconstruct the path when a client ask it.

To determine the reachable points according to the level of the skier, I used the Depth First Search algorithm. I had to modify it a little bit to handle the level of the skier but it work's fine.

4 Example

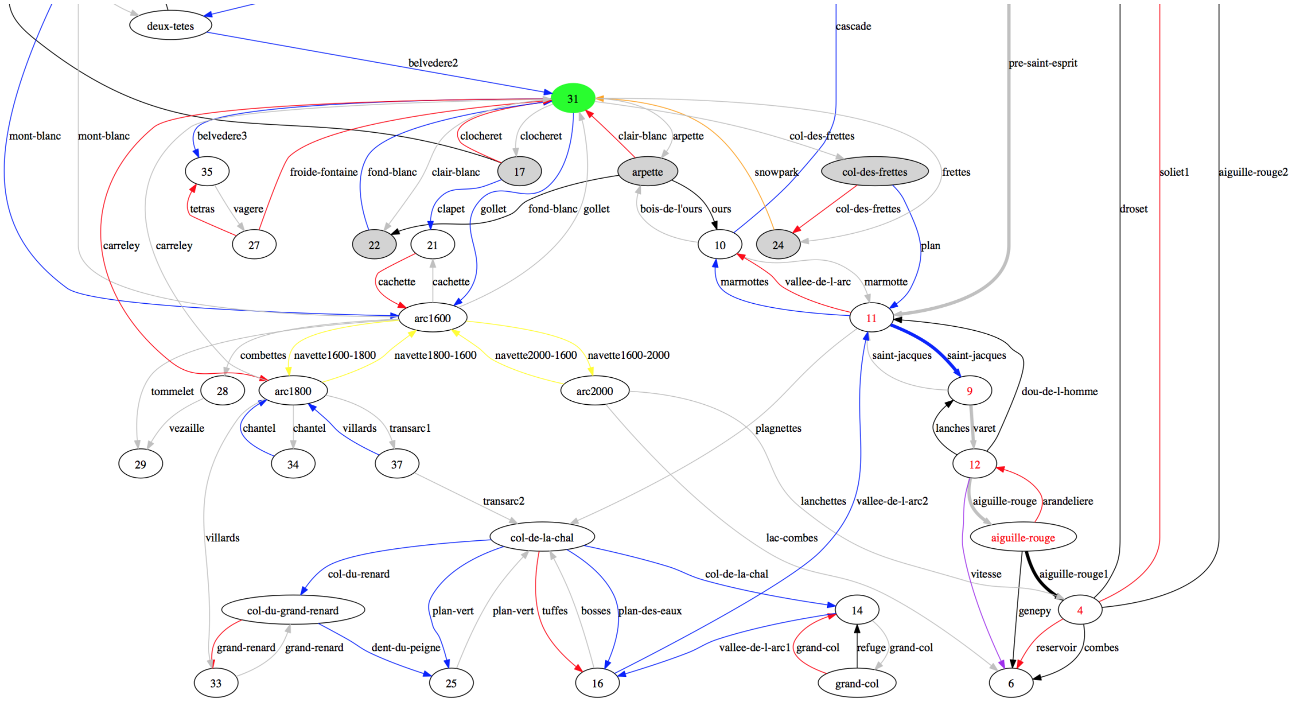
Here is an example for the shortest path between 1 and 4 and the reachable points from node 31 with only green routes.



```
12:52:22 goberle@asgard ~/Documents/AG44/SkiResort$ python skiresort.py -h
usage: skiresort.py [-h] [-f F] [-o O] s d rp lvl

AG44 : SkiResort

positional arguments:
  s           Source node for shortest path.
  d           Destination node for shortest path.
  rp          Source node for reachables points
  lvl         List of routes depending on the level of the skier.
              (V,B,R,N,KL,SURF)

optional arguments:
  -h, --help  show this help message and exit
  -f F        Data file
  -o O        Output file for the graph
12:52:26 goberle@asgard ~/Documents/AG44/SkiResort$ python skiresort.py 1 4 31 V
Shortest Path : [1, 2, 3, 8, 11, 9, 12, 5, 4] in : 7189.2seconds
Reachable Points : [31, 15, 17, 22, 23, 24]
12:52:31 goberle@asgard ~/Documents/AG44/SkiResort$
```

5 Graph

In order to have a big picture of what's the routes look like, I have done a Graph with the graphviz library. For the shortestpath, nodes fontcolor are in red and their edges are heavier and for the rechable points problem, the start node is in green and the reachable points are in grey. Here is a screenshot of the graph :



deux-tetes
cascade
belvedere2
pre-saint-esprit
31
mont-blanc
mont-blanc
belvedere3
clocheret
clocheret
clair-blanc
arpette
col-des-frettes
soliet1
aiguille-rouge2
35
froide-fontaine
fond-blanc
clair-blanc
17
arpette
snowpark
col-des-frettes
frettes
droset
tetras
vagere
clapet
gollet
fond-blanc
gollet
bois-de-l'ours
ours
col-des-frettes
carreley
carreley
27
22
21
10
24
plan
cachette
cachette
marmottes
vallee-de-l-arc
marmotte
arc1600
11
combettes
navette1600-1800
navette1800-1600
navette2000-1600
navette1600-2000
saint-jacques
saint-jacques
tommelet
28
arc1800
arc2000
9
dou-de-l-homme
vezaille
chantel
chantel
villards
transarc1
plagnettes
lanches
varet
29
34
37
12
transarc2
lanchettes
vallee-de-l-arc2
aiguille-rouge
arandeliere
villards
col-de-la-chal
lac-combes
aiguille-rouge
col-du-renard
col-de-la-chal
vitesse
aiguille-rouge1
col-du-grand-renard
plan-vert
plan-vert
tuffes
bosses
plan-des-eaux
14
genepy
4
grand-renard
grand-renard
dent-du-peigne
vallee-de-l-arc1
grand-col
refuge
grand-col
reservoir
combes
33
25
16
grand-col
6